

**AMENDMENTS TO THE SPECIFICATION**

**Please replace the paragraph beginning on Page 3, line 26 and ending on Page 4, line 2 with the following amended paragraph:**

Further, the second message may contain a group identifier. The first network switch may read the group identifier and generate a new group identifier as a function of the group identifier of the second message and the interface identifier data of the interface at which the first network switch received the second message. The several ~~SCVs~~ SVCs released by the first network switch may correspond to the new group identifier. Lastly, the first network switch may replace the group identifier in the second message with the new group identifier thereby creating a first modified second message. This first network switch may output the modified second message at another of its interfaces.

**Please replace the paragraph on Page 4, lines 4 through 7 with the following amended paragraph:**

The present invention may be better understood, and its ~~numbers~~ numerous objects, features and advantages made apparent to those skilled in the art by referencing the accompanying drawings. The use of the same reference number throughout the several Figures designates a like or similar element.

**Please replace the paragraph on Page 6, lines 8 through 19 with the following amended paragraph:**

Before data packets of a call between, for example end devices 242 and 244 can be transmitted, a data path must be established within ATM network 200. When establishing this data path, switches receive setup messages instructing them to allocate portions of their resources for SVCs. For example, switch 206 may receive a setup message for setting up a call between end devices 242 and 244. Figure 4A illustrates in block diagram form an exemplary setup message 400 received by switch 206. Setup

message 400 includes several identification element (IE) fields IE1 through IEX in which setup message data is stored. First field IE1 identifies message 400 as a setup message. Second field IE2 of message 400 contains a call reference unique to the call to be established between end device 244 and end device 242. In the exemplary embodiment shown in Figure 4A, the call reference is identified as C1. Fields IE3 and IE4 store a data transit list and group identifier, respectively, as will be more ~~full~~ fully described below.

**Please replace paragraph the paragraph beginning on Page 6, line 27 and ending on Page 7, line 10 with the following amended paragraph:**

Since switch 206 is the first switch to receive the setup message 400, switch 206 calculates a data transit list (DTL) for the call. The DTL defines the path in terms of the switches within network 200 through which data packets of call C1 will be transmitted. The DTL is dynamically calculated based upon the current state of network 200 in a ~~matter~~ manner which most efficiently transmits data packets of call C1 through network 200. For example, switch 206 may calculate a most efficient data path between end devices 244 and 242 to include switches 202, 204, and 206. The DTL of this path may be defined as ~~[[DBA]]~~ CBA where ~~[[D]]~~ C represents switch 206, B represents switches 204, and A represents switch 202. In this representative data path, data packets originating with end device 244 travel sequentially through switches 206, 204, and 202 before arriving at end device 242. In contrast, data packets originating from end device 242 travel sequentially through switch 202, switch 204, and switch 206 before arriving at end device 244. Switch 206 may map DTLs to corresponding calls in memory 304. Figure 3B shows map 312 that maps several calls to respective DTLs calculated by switch 206 or other switches including the DTL of the exemplary call C1. Switch 206 will insert the calculated DTL into IE3 of setup message 400 to create modified setup message 402 as shown in Figure 4B.

**Please replace the paragraph on Page 7, lines 11 through 19 with the following amended paragraph:**

Switch 206 may also calculate a group identifier in response to receiving the setup message 400. In general, the group identifier generated by switch 206 can be original or it can be a modification of a group identifier contained in IE4 of the received setup message. The group identifier may be a concatenation of the IDs of the switch interfaces through which switches receive a setup message when establishing a data path for a particular call. In the currently described embodiment, IE4 of the setup message 400 received by switch 206 lacks a group identifier since switch 206 is the first switch to receive the setup message for establishing a data path for the example call C1. Switch 206, in this embodiment simply stores the interface ID 1 in ~~[[IE3]]~~ IE4 as an original group identifier.

**Please replace the paragraph on Page 8, lines 26 through 31 with the following amended paragraph:**

In response to receiving setup message 402 shown in Figure 4B, switch 204 adds call reference C1 to record 618 to produce modified tree structure 630 shown in Figure 6B. Additionally, switch 204 allocates an SVC to call C1. The SVC allocated for call C1 is mapped to C1 in the data memory of switch 204. Either before or after updating the topology tree 600, switch 204 transmits setup message 404 to switch 202 via link 212 in accordance with the DTL ~~“DBA”~~ “CBA” in IE3.

**Please replace the paragraph on Page 9, lines 1 through 15 with the following amended paragraph:**

Switch 202, in response to receiving the setup message 404 at interface 2, allocates resources and to create a unique SVC. The SVC resources allocated for call C1 are mapped to C1 in the data memory of switch 202. Also, switch 202 updates the tree

topology stored in data memory thereof. Figure 7A represents the topology tree 700 stored in the data memory of switch 202 when switch 202 receives setup message 404. Topology tree 700 includes nodes 702 through 720 in addition to records 722 through 728. It is noted that records 722 through 728 correspond to nodes 704, 716, 718, and 728, respectively. Record 722 contains call references CX, CY, and CZ. Record 724 includes call reference CZ. Record ~~[[728]]~~ 726 includes call reference CY. Record 728 includes call reference CX. It is noted that call reference CY corresponds to a call with a DTL assigned as “ACE.” Root node 702 corresponds to switch 202 while nodes 704 through 708 correspond to interfaces of switch 202 designated 1, 2, and 3 respectively. Node 710 corresponds to interface 4 of switch 204, node 712 corresponds to interface 5 of switch 208, node 714 corresponds to interface 3 of switch 206, node 716 corresponds to interface 1 of switch 206, node 718 corresponds to interface 2 of switch 210, and node 720 corresponds to interface 2 of switch 210.

**Please replace paragraph the paragraph on Page 10, lines 14 through 23 with the following amended paragraph:**

In the currently described embodiment, IE4 of the connect message 800 received by switch 202 lacks a group identifier since switch 202 is the first switch to receive the connect message for acknowledging the data path setup between end devices 242 and 244. Switch 202, in this embodiment simply stores the interface ID 1 in ~~[[IE3]]~~ IE4 as an original group identifier thereby producing modified connect message 802 shown in Figure 8B. Additionally, switch 202 updates its topology tree stored in its data memory. More particularly, switch 202 adds the call reference of connect message 800 to record 722 shown in Figure 7B to produce tree 740 shown in Figure 7C. Eventually switch 202 transmits the updated connect message 802 shown in Figure 8B to switch 204 via link 212 in accordance with the reverse order of the DTL contained in IE3.

**Please replace paragraph the paragraph beginning on Page 11, line 27 and ending on Page 12, line 2 with the following amended paragraph:**

To illustrate, suppose switch 206 detects a failure of link ~~[[202]]~~ 222 through which data packets of call C1 and other calls are being transmitted. Switch 206 detects the failure and in response, switch 206 may generate the restart message 900 shown in Figure 9A. Restart message 900 includes fields designated by IE1 through IEX. Field IE1 indicates that message 900 is a restart message. Switch 206 stores in field IE2 the interface ID of the interface coupled to the failed link. In this case, switch 206 stores interface ID 1 in IE2.

**Please replace paragraph the paragraph on Page 13, lines 27 through 31 with the following amended paragraph:**

Although the present invention ~~have~~ has been described in connection with several embodiments, the invention is not intended to be limited to the specific forms set forth herein, but on the contrary, it is intended to cover such alternatives, modifications, and equivalents as can be reasonably included with in the spirit and scope of the invention as defined by the appended claims.